Solution

Process Interaction

- Unaware of each other
  - Independent processes
  - Not intended to work together
    - Example: Netscape and Word
  - Compete for resources
    - Printer, files, CPU time
  - O.S. must manage mutual exclusion, avoid deadlock and starvation
- Indirectly aware of each other
  - Share read/write access to a object
  - Example: Word and Excel
  - Cooperate in using the shared object
  - Must also worry about data coherence ← ترابط
- Directly aware of each other
  - Designed to work together, communicate
  - Example: Background printing process that is part of a word processing program

Q4-a:- give an example of a deadlock between two process?

Solution:-

Department of Electrical and Electronic Engineering
Faculty of Engineering
Alfatah University

Test: *First Midterm Examination.* **Semester:** *Spring 2006.*
Course Code: *EE433* **Course Name:** *Operating Systems*
Time allowed: *1.5 Hour* **Maximum marks:** *20.*

Answer all Questions.

1- (a) Sketch a *five* state model for the *process* and write these states
(b) Write the necessary *parts* of information the *system* has to keep to control a *process.*
(c) What are the *steps* of creating a *process.*

2- (a) What is a *Thread*?
(b) Name *four* Thread *operations.*
(c) How many *relations* between *Threads* and *Processes*? Explain each of them.

3- (a) Define the following terms:
i- *Mutual Exclusion.*
ii- *Critical Section.*
iii- *Deadlock.*
iv- *Starvation*

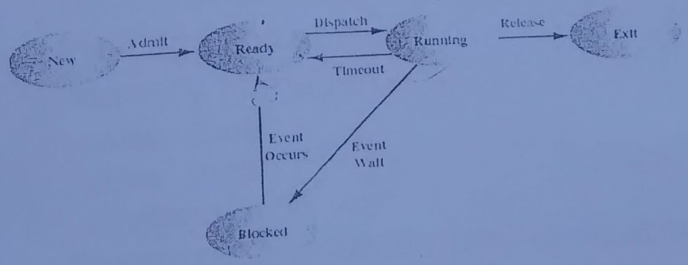
(b) In process Interaction, processes are either *unaware* of each other, *indirectly* aware of each other or *directly* aware of each other. Explain with examples these *three* cases.

4- (a) Give an example of a *deadlock* between *two* process.
(b) What are the *reusable* resources and the *consumable* resources?
(c) Name *four* conditions for *Deadlock* and explain each of them.

Q1-a:- sketch a five state model for the process and write state ?

Solution :-

Five-state Model



Five-State Process Model

Five-state Model

- Processes may be waiting for I/O
- Use additional states:
  - *Running*: currently being run
  - *Ready*: ready to run
  - *Blocked*: waiting for an event (I/O)
  - *New*: just created, not yet admitted to set of runnable processes
  - *Exit*: completed/error exit

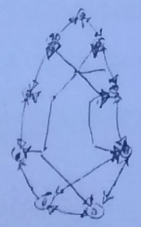
انتهت إجابة السؤال الأول للفقرة A

Q1-b:-write the necessary parts of information the system has to keep to control a presser ?

Solution:-

- Process Control Information
  - Scheduling/state Priority
  - Data structure info (links)
  - Interprocess Communication
  - Privilege Information
  - Memory Management
    - User Code
    - User Data
    - System Stack
  - Resource Ownership and Utilization

انتهت إجابة السؤال الأول للفقرة B



Q1-c:- What are the steps of creating a process?

Solution :-

Creating a process

1. Assign a unique process ID
2. Allocate memory for the process
   - Can use defaults for amount of memory
   - May get values from creating process
   - Link to shared memory (clipboard)
3. Initialize the process control block
   - Process ID, parent ID
   - Most of this information can be zeroed
     - Set program counter and stack pointer to appropriate values
     - May set up argc/argv information
   - State usually set to Ready
4. Set links so it is in the appropriate queue
5. Create other data structures

Memory - files - accounting.

Q2-a:-what is a thread?

Solution:-

A thread is basic unit of CPU utilization ;it comprises a thread ID,a program counter , a register set ,and a stack

Q2-b:-Name four thread operations ?

Solution:-

- Thread operations
  - *Spawn* – Creating a new thread
  - *Block* – Waiting for an event
  - *Unblock* – Event happened, start new
  - *Finish* – This thread is completed

Q2-c:-how many relations between third and processes? Explain each of them ?

Solution:-

Threads and Processes

- 1 to 1
  - Each process has one thread (Unix)
- Many to 1
  - Each process can have multiple threads (NT, Solaris, OS/2, Mach)
  - Very common
- 1 to Many
  - Seen in some distributed operating systems
  - Thread can move between machines and address spaces
- Many to Many
  - Can switch to a different domain (I/O) to perform key operations
  - Helps create a structured program

Q3-a:-Define the following terms :

i-mutual Exclusion.

ii-critical section.

iii-Deadlock.

iv-starvation.

Solution:-

Definitions

- *Mutual Exclusion* – Making sure two processes can't both have a resource
- *Critical Section* – The area of a program where a resource is being used
- *Deadlock* – When two or more processes halt, unable to proceed
  - P1 is using A, needs B
  - P2 is using B, needs A
- *Starvation* – A process waits indefinitely for a resource
  - P1 using A, P2 and P3 wait for A
  - P2 gets A when P1 done
  - P1 comes in, P1 and P3 wait for A

Q3-b:-

In process interaction ,processes are either unaware of each other indirectly aware of each other or directly aware of each other .Explain with examples these three cases ?

Solution

Process Interaction

- Unaware of each other
  - Independent processes
  - Not intended to work together
    - Example: Netscape and Word
  - Compete for resources
    - Printer, files, CPU time
  - O.S. must manage mutual exclusion, avoid deadlock and starvation
- Indirectly aware of each other
  - Share read/write access to an object
  - Example: Word and Excel
  - Cooperate in using the shared object
  - Must also worry about data coherence
- Directly aware of each other
  - Designed to work together, communicate
  - Example: Background printing process that is part of a word processing program

Q4-a:- give an example of a deadlock between two process?

Solution:-

- Deadlock Example

Process P	Process Q
(1) Get A	(5) Get B
(2) Get B	(6) Get A
(3) Release B	(7) Release A
(4) Release A	(8) Release B

- What is deadlock?
  - A group of processes permanently stuck waiting for each other

Q4-b:-
What are the reusable resources and the consumable resources?
Solution:-

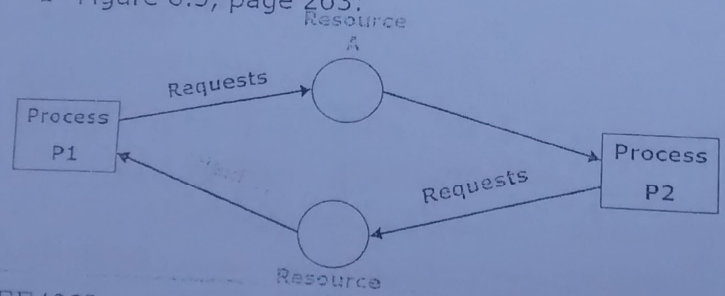
- Reusable Resources
  - Resources (printer, memory, files, etc.) can only be used by one process at a time. After use, can be reassigned to another process.
  - In previous example, if "A" = file, and "B" = tape drive, deadlock can occur
  - Can occur with memory allocation if we don't do swapping
    - 200K available, both processes want 80K, then make second request for 70K

Q4-c:- name four conditions for deadlock and explain each of them?
Solution:-

Conditions for Deadlock

- All must hold for deadlock to occur

1. Mutual Exclusion
   - Only one process can hold the resource at any given time
2. Hold and Wait
   - A process may hold resources while requesting others
3. No Preemption
   - No resource may be forcibly removed from a process
4. Circular Wait (event)
   - A closed chain of process exists, each waiting for a resource held by the next process in the chain
   - Figure 6.5, page 263:



Explain th O.S. Kernel structure?

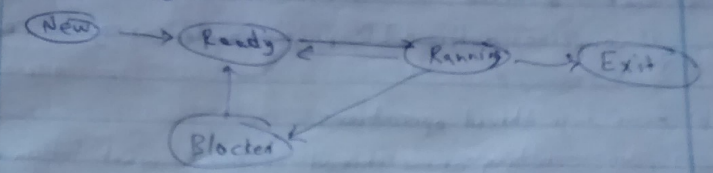
- Non process Kernel
- Kernel runs outside the context of any running process
- Kernel has its own memory and stack

2- Execution within user Process
- system calls occur in the context of the current user process
- common in minicomputers and Micro computers
- Each process image has both user and system data structures

3- Process based O.S
- Implements the O.S as a collection of system processes
- Encourages a clean design
- useful in a multiprocessor system

1

a) five state model



New: just created not yet admitted to set of Runnable proces
Running ←, Blocked: waiting for I/o event
Ready →, Exit: completed / error exit

b) process control information
- scheduling / state priority
- Data structures info (Links)
- Interprocess communication
- privilege info
- Memory management → user code, data, system stack
- Resource ownership & utilization

c) Steps of creating a process

1- assign a unique process ID

2- Allocating memory for process
→ can use default amount of memory
→ May get values from creating process
→ Links to shared memory (clipboard)

3- Initialize PCB
→ process ID / parent ID
→ most values can be zeroed → May set args / args

②

a) what is a thread? it is a basic unit of CPU utilization
it comprises a thread ID, program counter, register set, stack

b) name four thread operations:
1) Spawn: creating anew thread 2) block: wait for an event
3) Unblock: event happened 4) finish: thread completed

c) explain Process-thread relation?
1) 1 to 1: each process has one thread (unix)
2) Many to 1: each process contains multiple threads (Core, Mach, Sola)
very common
3) Many to Many: can switch to different domain (I/O) to perform key operations - helps create a structured program
4) 1 to many: - seen in some distributed OS
- Thread can move between machines & address spaces

Q3) define:
Mutual exclusion: Making sure two process can't both have the same resource

critical section: an area of a program where a resource is used

Deadlock: when two processes halt, unable to proceed (A vs. B)

Starvation: a process waits indefinetly for a resource (P1 vs P2)

3B) Process interaction
- Unaware of each other:
  - independent processes - not intended to work together (Firefox & word)
  - compete for resources
  - OS must avoid deadlock & starvation, implement mutual exclusion
- Indirectly aware of each other:
  - share read/write access to object (word & excel)
  - Cooperate in using shared object
  - Must worry about data coherence
- Directly aware of each other
  - Designed to work & communicate with each other
  - example: Background printing process that is a part of whole program

Q4) give an example of deadlock
a)

Process P	Process Q
1) Get A	5) Get B
2) Get B	6) Get A
3) Release B	7) Release A
4) Release A	8) Release B

- Deadlock: a group of processes permanently stuck waiting for each other

c) four conditions for deadlock: All must hold for deadlock to occ
1) Mutual exclusion: only one process can hold a resource at any tim
2) Hold & wait: a process may hold a resource while requesting anoth
3) No preemption: No resource may be forcibly removed from proces

Explain Kernel structure:

1) Non process Kernel:
- Kernel runs outside the context of any running process
- Kernel has its own memory & stack.

2) Execution within user process:
- system calls occur within the context of current user process
- Common in mini & microcomputers.
- Each process image has both user & system data structures

3) process based O.S.:
- implement OS as collection of system processes
- encourages a clean design
- useful in multiprocessor system.

Maximum marks: 20

Answer all Questions.

Name: ID:

1- (a) Define the following terms: (2 marks)
i. Process.
ii. Kernel of an OS.

(b) Explain with aid of figures the idea of multiprogramming. (2 marks)

2- (a) Sketch a *five* state model for the *process* and write these states. (1.5 mark)

(b) Write the necessary *parts* of information the *system* has to keep to control a *process*. (2 marks)

(c) What are the *steps* of creating a *process*. (2.5 marks)

3- (a) What is a *Thread*? (1 mark)

(b) Name *four* Thread *operations*. (2 marks)

(c) How many *relations* between *Threads* and *Processes*? Explain each of them. (2 marks)

4- (a) Define the following terms: (2 marks)
i- *Mutual Exclusion.*
ii- *Critical Section.*
iii- *Deadlock.*
iv- *Starvation.*

(b) In process interaction, processes are either *unaware* of each other, *indirectly* aware of each other or *directly* aware of each other. Explain with examples these *three* cases. (3 marks)

Good luck

Dr. Abdalla Abushafa - 2016

Department of Electrical and Electronic Engineering
Faculty of Engineering
Alfatah University

Test: *First Midterm Examination.* Semester: *Spring 2006.*
Course Code: *EE433* Course Name: *Operating Systems*
Time allowed: *1.5 Hour* Maximum marks: *20.*

Answer all Questions.

1- (a) Sketch a *five* state model for the *process* and write these states
(b) Write the necessary *parts* of infonnation the *system* has to keep to control a *process.*
(c) What are the *steps* of creating a *process.*

2- (a) What is a *Thread?* unit of execution can used to dispatch
(b) Name *four* Thread *operations.*
(c) How many *relations* between *Threads* and *Processes*? Explain each of them.

3- (a) Define the following terms:
i- *Mutual Exclusion.*
ii- *Critical Section.*
iii- *Deadlock.*
iv- *Starvation*

(b) In process Interaction, processes are either *unaware* of each other, *indirectly* aware of each other or *directly* aware of each other. Explain with examples these *three* cases.

4- (a) Give an example of a *deadlock* between *two* process.
(b) What are the *reusable* resources and the *consumable* resources?
(c) Name *four* conditions for *Deadlock* and explain each of them.

Cn=3Cc+Cs
Cb=3Cs
Ca=2Cc+Cs

Cn=3Cc+Cs

Cb=3Cs

Ca=2Cc+Cs

Dr. Abubaker Abushofa - 2006

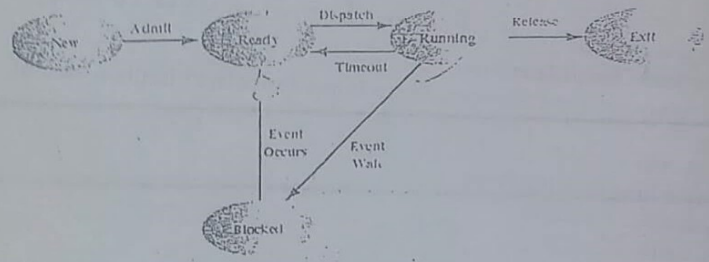
Good luck

10 ∑Cn=

Q1-a:- sketch a five state model for the process and write state ?

Solution :-

Five-state Model



Five-State Process Model

Five-state Model

- Processes may be waiting for I/O
- Use additional states:
  - *Running*: currently being run
  - *Ready*: ready to run
  - *Blocked*: waiting for an event (I/O)
  - *New*: just created, not yet admitted to set of runnable processes
  - *Exit*: completed/error exit

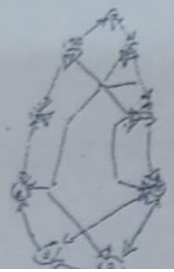
انتهت إجابة السؤال الأول للفقرة A

Q1-b:-write the necessary parts of information the system has to keep to control a presser ?

Solution:-

- Process Control Information
  - Scheduling/state, Priority
  - Data structure info (links)
  - Interprocess Communication
  - Privilege Information
  - Memory Management
    - User Code
    - User Data
    - System Stack
  - Resource Ownership and Utilization

انتهت إجابة السؤال الأول للفقرة B



Q1-c:- What are the steps of creating a process ?

Solution :-

Creating a process

1. Assign a unique process ID
2. Allocate memory for the process
   - Can use defaults for amount of memory
   - May get values from creating process
   - Link to shared memory (clipboard)
3. Initialize the process control block
   - Process ID, parent ID
   - Most of this information can be zeroed
     - Set program counter and stack pointer to appropriate values
     - May set up argc/argv information
   - State usually set to Ready
4. Set links so it is in the appropriate queue
5. Create other data structures

Memory - files - accounting

Q2-a:-what is a thread?

Solution:-

A thread is basic unit of CPU utilization ;it comprises a thread ID, a program counter , a register set ,and a stack

Q2-b:-Name four thread operations ?

Solution:-

- Thread operations
  - *Spawn* – Creating a new thread
  - *Block* – Waiting for an event
  - *Unblock* – Event happened, start new
  - *Finish* – This thread is completed

Q2-c:-how many relations between third and processes? Explain each of them ?

Solution:-

Threads and Processes

- 1 to 1
  - Each process has one thread (Unix)
- Many to 1
  - Each process can have multiple threads (NT, Solaris, OS/2, Mach)
  - Very common
- 1 to Many
  - Seen in some distributed operating systems
  - Thread can move between machines and address spaces
- Many to Many
  - Can switch to a different domain (I/O) to perform key operations
  - Helps create a structured program

Q3-a:-Define the following terms :

i-mutual Exclusion.

ii-critical section.

iii-Deadlock.

iv-starvation.

Definitions

- *Mutual Exclusion* – Making sure two processes can't both have a resource
- *Critical Section* – The area of a program where a resource is being used
- *Deadlock* – When two or more processes halt, unable to proceed
  - P1 is using A, needs B
  - P2 is using B, needs A
- *Starvation* – A process waits indefinitely for a resource
  - P1 using A, P2 and P3 wait for A
  - P2 gets A when P1 done
  - P1 comes in, P1 and P3 wait for A

Q3-b:-

In process interaction ,processes are either unaware of each other indirectly aware of each other or directly aware of each other .Explain with examples these three cases ?

7